

AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph on page 10, lines 13-26 with the following amended paragraph:

In accordance with yet another embodiment, the computing system 100 further comprises an operating system and usually one or more application programs. Such an embodiment is familiar to those of ordinary skill in the art. The operating system comprises a set of programs that control operations of the computing system 100 and allocation of resources. The set of programs, inclusive of certain utility programs, also provide a graphical user interface to the user. An application program is software that runs on top of the operating system software and uses computer resources made available through the operating system to perform application specific tasks desired by the user. In accordance with an embodiment, the operating system employs a graphical user ~~interface 100~~ interface wherein the display output of an application program is presented in a rectangular area on the screen of the display device 114. The operating system is operable to multitask, i.e., execute computing tasks in multiple threads, and thus may be any of the following: Microsoft Corporation's "WINDOWS 95," "WINDOWS CE," "WINDOWS 98," "WINDOWS 2000" or "WINDOWS NT" operating systems, IBM's OS/2 WARP, Apple's MACINTOSH OSX operating system, Linux, UNIX, etc.

Please replace the paragraph on page 15, lines 5-15 with the following amended paragraph:

From the foregoing, it should be understood that the customization program 300 is used to create source code for the configuration file 306 as well as model the configuration of the baseboard relevant to the BMC 104. As such, the initial design page 406 is blank and void of any component icons (e.g. 402). In order to build the source code for this configuration file 306, the configuration routine 312 first detects on which addresses of the management bus 130 a component is directly or indirectly connected. Then, the configuration routine 312 builds a list of addresses on which a component is detected. Next, the configuration routine 312 analyzes each DDF 304 stored in the DDF library 310 to ~~identification~~ identify information associated with the component accessible on each address. In an embodiment, this analysis is accomplished

based on commands sent to the components by the program 300 and subsequent responses sent by the detected components and including the identification information.

Please replace the paragraph on page 22, lines 19-23 with the following amended paragraph:

The second query operation 808 determines whether all slave addresses accessible to the BMC 104, either directly or by way of a sensor aggregation device 202 have been pinged by the ping operation ~~804~~ 804. If so the operation flow concludes at the terminate operation 812. Otherwise, if all slave addresses have not been pinged by the ping operation 804, the operation flow is passed from the second query operation 808 back to the ping operation 804.

Please replace the paragraph on page 23, lines 12-20 with the following amended paragraph:

The execute operation 908 executes the query to determine whether the component detected on the extracted slave address is the type of component corresponding to the extracted DDF 304. Upon execution, the query issues one or more commands to the component detected on the extracted slave address. These command(s) are specific to, and thus, only recognizable to, the component corresponding to the DDF 304, based, for example, on classification and type. As such, upon reception of the command(s), this specific component is caused to respond in such a manner as to be identified on the extracted slave address. In contrast, all other components do not recognize these one or more commands, and thus, do not respond. To that end, the operation flow passes to the first query operation 910 after the execute operation 908 has been performed.